

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-234040

(43)Date of publication of application : 02.09.1998

(51)Int.Cl.

HO4N 7/32
// HO4L 12/56

(21)Application number : 10-027233

(71)Applicant : TEXAS INSTR INC <CTI>

(22)Date of filing : 09.02.1998

(72)Inventor : LAGENDLA K TARURI
JEAN CHEUN

(30)Priority

Priority number : 97 37729 Priority date : 07.02.1997 Priority country : US

(54) METHOD FOR PACKETIZING VIDEO INFORMATION

(57)Abstract:

PROBLEM TO BE SOLVED: To use motion vector data even under the presence of an incorrectable error by dividing motion vector data of each of a plurality of pixel groups and corresponding texture data so as to insert a re-synchronization word to a video bit stream.

SOLUTION: A frame is divided into blocks or macro blocks MB, and most MBs are coded by a motion vector MV for prediction from a preceding MB and texture data DCT corresponding to the difference between a predicted MB pixel and an actual MB pixel. A motion re-synchronization word is selected from a possible word found out through the retrieval based on a corresponding variable length coding(VLC). The re-synchronization word assist error check even when some of data are to be aborted so as to attain the use of partial data, and even when an incorrectable error is in existence in a packet, partial reproduction is attained by the addition of an overhead.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-234040

(43)公開日 平成10年(1998)9月2日

(51)Int.Cl.⁶

H 04 N 7/32

// H 04 L 12/56

識別記号

F I

H 04 N 7/137

Z

H 04 L 11/20

102 F

審査請求 未請求 請求項の数 2 OL (全 10 頁)

(21)出願番号 特願平10-27233

(22)出願日 平成10年(1998)2月9日

(31)優先権主張番号 037729

(32)優先日 1997年2月7日

(33)優先権主張国 米国 (US)

(71)出願人 590000879

テキサス インスツルメンツ インコーポ
レイテッド

アメリカ合衆国テキサス州ダラス, ノース
セントラルエクスプレスウェイ 13500

(72)発明者 ラジェンドラ ケイ. タルリ

アメリカ合衆国 テキサス州プラノ, ファ
ウンテン ヘッド ドライブ 2220

(72)発明者 ジーン チェウン

アメリカ合衆国 カリフォルニア州バーク
レイ, サンタ クララ アベニュー 560

(74)代理人 弁理士 浅村 皓 (外3名)

(54)【発明の名称】 ビデオ情報をパケット化する方法

(57)【要約】

【課題】 いくらかの訂正不可能なエラーが存在しても、動きベクトルデータがなお使用可能であるようにし、圧縮ビデオデータのパケット内に訂正不可能なエラーある場合でも、わずかなオーバヘッドの追加により部分再生を可能にする。

【解決手段】 (マクロ) ブロックレベルの動き補償により圧縮されたビデオのビットストリームにおいて、動きベクトルは集約されて、対応するテクスチャデータから、再同期ワードにより分離される。また、可変長コードテーブルから再同期ワードを発生する方法が提供され、該方法により発生せしめられた再同期ワードを用いて、該再同期ワードに隣接する動きベクトルまたはテクスチャデータがコード化される。

| スライス 再同期 | MB 番号 | 0 | MV _n | MV _{n+1} | ... | MV _{n+k} | 動き 再同期 | DCT _n | ... | DCT _{n+k} |
|-------------|----------|---|-----------------|-------------------|-----|-------------------|-----------|------------------|-----|--------------------|
|-------------|----------|---|-----------------|-------------------|-----|-------------------|-----------|------------------|-----|--------------------|

【特許請求の範囲】

- 【請求項1】 (a) 対象についての複数のピクセルグループの各々の動きデータおよびテクスチャデータを発生するステップと、
 (b) 前記動きデータを集約するステップと、
 (c) 前記テクスチャデータを集約するステップと、
 (d) 前記集約された動きデータと、前記集約されたテクスチャデータとの間に、再同期ワードを挿入するステップと、を含む、ビデオ情報をパケット化する方法

【請求項2】 動き補償されたビデオビットストリームシンタックスであって、

- (a) ビットストリーム内の第1グループの連続ビットであって、少なくとも2つの動きベクトルをコード化した該第1グループの連続ビットと、
 (b) 前記ビットストリーム内の前記第1グループのビットに続く第2グループの連続ビットであって、再同期ワードを形成する前記第2グループの連続ビットと、
 (c) 前記第2グループに続く第3グループの連続ビットであって、前記動きベクトルに関連するテクスチャデータをコード化した前記第3グループの連続ビットと、
 を含む、前記動き補償されたビデオビットストリームシンタックス。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、雑音性 (noisy) チャネルおよび記憶装置を経ての伝送のための情報のコード化に関し、特にエラー許容性 (resilient) コード化に関する。

【0002】

【従来の技術】 雑音性チャネルを経てのデータの送信中に生じるエラーを軽減するためには、2つの一般的なアプローチが存在する。すなわち、自動再送要求 (Automatic Retransmission) (ARQ) およびフォワードエラーコレクション (Forward Error Correction) (FEC) である。ARQタイプの軽減は、ビデオのようなマルチキャストまたはリアルタイムのアプリケーションにおいては、許容しない時間遅延、または帰還チャネルの欠如のために適さない。そのような場合には、デコーダーは、ある程度までエラー訂正コード化により保護された、エラーにより劣化したビットストリームのみをデコードでき、そのようなビットストリームから作成を行わなければならない。FECは、エラー訂正コード (例えば、リードソロモン) による軽減を行う。しかし、訂正不可能なエラーは、さらなる軽減アプローチを必要とする。

【0003】 一般に、通常用いられるビデオ圧縮方法は、時間的冗長性を除去するためにブロックに基づく動き (motion) 補償を行う。動き補償方法は、(マクロ) ブロック動きベクトルおよび量子化残差 (res

iduals) (テクスチャ) のみをコード化し、動きベクトルおよび残差の可変長コーディング (VLC) はコード化効率を増大させる。しかし、可変長コーディングは、しばしば送信チャネルエラーの影響を極めて受けやすく、デコーダは、訂正不可能なエラーが発生した時、エンコーダとの同期を容易に失う。動き補償のような予測コーディング方法は、事態を著しく悪化させる。そのわけは、1つのビデオフレーム内のエラーが、全ビデオシーケンスを経て高速で伝搬し、デコードされるビデオの品質を高速で劣化させるからである。

【0004】 訂正不可能なエラーに対する、そのようなブロックに基づくビデオ圧縮方法の典型的なアプローチは、エラー検出 (例えば、範囲外動きベクトル、無効な VLC テーブルエントリ、またはブロック内の残差の無効な数) のステップと、デコーダのエンコーダとの再同期のステップと、訂正不可能なデータの代わりに前に送信された訂正されたデータの繰返しにより行うエラー補正のステップと、を含む。例えば、MPEG 1-2 を用いて圧縮されたビデオは、フレームのマクロブロック (MB) のそれぞれのスライスの開始点に再同期マーカ (開始コード) を有し、訂正不可能なエラーは、正しくデコードされた再同期マーカ間の全てのデータを廃棄されることになる。これは、殊に MPEG のような予測圧縮方法における、ビデオストリームの品質の劣化を意味する。

【0005】 これらのビデオ圧縮および圧縮解除の方法は、特殊な集積回路において、またはプログラム可能デジタル信号プロセッサ、またはマイクロプロセッサにおいて実行されうる。

【0006】

【発明が解決しようとする課題】 本発明の目的は、いくらかの訂正不可能なエラーが存在していても、動きベクトルデータがなお使用可能であるようにし、圧縮ビデオデータのパケット内に訂正不可能なエラーある場合でも、わずかなオーバヘッドの追加により部分再生を可能にすることである。

【0007】

【課題を解決するための手段】 本発明は、動きベクトルデータおよび対応するテクスチャデータを区分することにより、ビデオビットストリーム内に組み込まれた再同期を提供し、それによりいくらかの訂正不可能なエラーが存在していても、動きベクトルデータがなお使用可能であるようにする。本発明はまた、データの可変長コードと両立しうる再同期マーカとして用いるためのワードを選択する方法を提供する。これは、圧縮ビデオデータのパケット内に訂正不可能なエラーある場合でも、わずかな追加のオーバヘッドによる部分再生を含む利点を与える。

【0008】

【発明の実施の形態】 図面はわかりやすくするために概

略的なものとしてある。

ビットストリームシンタックス

図1は、動き再同期を有する第1実施例のビットストリームパケットシンタックスを示し、比較のために図2は、実施例の動き再同期のない、公知のパケットシンタックスを示す。特に、図1および図2は、MPEGタイプのコード化ビデオにおけるパケットシンタックスを示し、それはフレームをブロックまたはマクロブロック(MB)に区分し、大部分のMBを、前のMBからの予測のための動きベクトル(MV)と、予測されたMBピクセルと実際のMBピクセルとの間の(圧縮された)差に対するテクスチャデータ(DCT)と、によりコード化している。実際、352ピクセル毎ラインの288ラインから成る(輝度のみの)フレームは、 18×22 アレイのMBを形成し、それぞれのMBは 16×16 アレイのピクセルである。送信または記憶のためには、そのようなフレームの396MBのシーケンスは、便利なサイズのパケットに分割される。例えば、パケットサイズが約1000バイトであり、平均的MBが約320ビット(例えば、動きベクトルのための20ビットおよびテクスチャのための300ビット)によりコード化される場合は、パケットはほぼ25MBを含む。従って、そのようなフレームは、送信のためにほぼ16パケットを必要とし、30フレーム毎秒のフレームレートは、ほぼ3.8Mbpsの送信を必要とする。低ビットレートの送信においては、テクスチャデータは著しく減少せしめられるので、動きベクトルデータは比較的に、より重要なとなる。

【0009】典型的には、フレームのMBはスライスにグループ化され、1つのスライスはフレームの單一行内の1つまたはそれ以上の連続するMBから成る。従って、前述の例においては、1つのスライスは1ないし2MBを含みうる。1つのスライス内には、マクロブロックデータが順次現れる。再同期は、スライス開始コードを用いて行われる。例えば、単純化されたスライスは、ヒューリスティックには次のようなものでありうる。

```
slice()
{
    slice_start_code
    macroblock_number
    quantizer_scale_code
    do {
        macroblock(j)
    } while(j < macroblock_count)
    next_start_code()
}
```

ただし、macroblock()関数は、マクロブロックに対する動きベクトルおよびテクスチャデータを与え、macroblock_numberは、スライス内の第1マクロブロックのフレームスキャンの番号であ

る。これは、図2に示されているシンタックスを与え、動きベクトルおよびテクスチャデータは、マクロブロックを基礎としてインタリーブされている。

【0010】これとは対照的に、実施例のスライス関数は、動きベクトルデータおよびテクスチャデータを分離された部分に区分し、かつ動き再同期ワード(motion_resynch)をこれらのデータ部分の間に挿入し、これはヒューリスティックには次のようなものでありうる。

```
10 slice()
{
    slice_start_code
    macroblock_number
    quantizer_scale_code
    do {
        motion_vector(j)
    } while(j < macroblock_count)
    motion_resynch
    do {
        block(j)
    } while(j < macroblock_count)
    next_start_code()
}
```

ただし、motion_vector()関数およびblock()関数は、それぞれマクロブロックに対する動きベクトルおよびテクスチャデータを与え、macroblock_numberは、第1マクロブロックの番号である。motion_resynchワードは、追加のエラー補正を以下のように行う。

【0011】motion_resynchワードは、動きベクトルデータの任意の許されるシーケンスから少なくとも1のハミング距離となるように、動きベクトルVLCテーブルから計算される。motion_resynchワードは、動きベクトルVLCコードワードストリームから一意的にデコード可能であり、動きベクトルデータの読み取りを、テクスチャデータの読み取りを始める前のどこで停止すべきかのデコーダ知識を与える。パケット内のマクロブロックの数は、デコーダがmotion_resynchワードに遭遇した後に暗黙的に知られる。デコーダはエラー検出に応答し、エラーのタイプに依存して動作を進める。

【0012】(1)訂正不可能なエラーが動きベクトルデータ内に検出された時は、デコーダはエラーのフラグを立て、次の再同期マーカまで、パケット内の全てのマクロブロックを、スキップされるブロックにより置換する。再同期は、次の満足に読み取られた再同期マーカにおいて行われる。もしいずれかの後のビデオパケットが、再同期の前に失われれば、それらのパケットはスキップされるマクロブロックにより置換される。

【0013】別のスキームにおいては、デコーダが動きベクトルデータ内にエラーを検出した時、デコーダは、

正しくデコードされた動きベクトルデータの部分（例えば「N」マクロブロック）を用いて動き補償を適用することをも選択しうる。デコーダは次に動きマーカを探索し、テクスチャデータの対応する「N」マクロブロックのデコードを試みる。もしこのテクスチャデータが検出可能なエラーなしにデコード可能であれば、デコーダは、このテクスチャデータをすでに動き補償された「N」ブロックに追加する。「N」マクロブロックテクスチャのいずれかのデコーディングにおいてもエラーが検出されれば、デコーダは、動き補償のみを用い、テクスチャデータをゼロで置換する。

【0014】(2) (motion_resynchワードに遭遇し、動きデータ内にエラーが検出されなかつた後に) テクスチャデータ内にエラーが検出された時は、動きベクトルデータは動き補償を行うために用いられる。マクロブロックのテクスチャデータは全てゼロにより置換され、デコーダは次の再同期マーカに再同期する。

【0015】(3) もしエラーが、動きベクトルデータまたはテクスチャデータ内に検出されないが、再同期マーカ1が、パケットの全てのマクロブロックのデコーディングの終了時に見出されなければ、エラーのフラグが立てられ、パケットのテクスチャデータのみが廃棄される。動きベクトルデータは検出されたmotion_resynchワードに先立って来ているため、動きベクトルデータには高い信頼性があるので、動き補償はなおマクロブロックに対して適用される。

【0016】(4) もしエラーが、現在のパケット内の動きベクトルデータまたはテクスチャデータ内に検出されず、次の再同期マーカが見出されれば、追加のチェックが行われる。すなわち、次のパケットの第1マクロブロックの番号から、現在のパケットの第1マクロブロックの番号を減算したものは、motion_resynchワードまでの動きベクトルデータのデコーディングにより見出された現在のパケット内のマクロブロックの数に等しくなければならない。もしこれら2つのマクロブロック数の測度が一致しなければ、この次のパケットのデータを廃棄する。そのわけは、この次のパケットの第1マクロブロックの番号にエラーがある可能性が高いからである。現在のパケットにエラーがある可能性は小さい。そのわけは、正しくデコードされた動きベクトルの数が、正しくデコードされたテクスチャデータアイテムの数に一致し、motion_resynchワードが正しい場所に見出されたからである。これは、図2のタイプのシンタックスとは対照的である。該シンタックスにおいては、そのようなエラーは、エラー発生の位置と、エラー検出の位置とが通常一致せず、いずれのパケットも信頼しえないので、両パケットが廃棄されることを要求する。

【0017】要するに、motion_resynch

ワードが与えるものは、(1) motion_resynchワードが動きベクトルデータの終わりに見出されなければならないことによる、動きベクトルデータの有効性のより厳しいチェックと、(2) 動きベクトルデータおよびテクスチャデータ内にエラーが検出されなかつたことと、次の再同期マクロブロックが見出されないこととが対になった時は、motion_resynchワードが正しく見出されているので、テクスチャデータの廃棄のみが要求されることと、である。

【0018】動き再同期ワードは、以下の節において説明するような探索を用い、動きVLCテーブルから計算されうる。このワードは、動きVLCテーブルから得られる任意の可能な有効な組合せから、少なくとも1のハミング距離にある。

【0019】再同期ワードの発生

第1実施例の方法は、対応するVLCテーブルに基づく探索により、再同期ワード（前節の動き再同期ワードは1つの例である）を作る。最適の再同期ワードは、その探索により見出された可能なワードから選択されうる。特に、ビットストリームはコードワード c_i^k （第k VLCテーブルからの第iコードワード）のシーケンスを有するものと仮定し、実施例の方法は、このビットストリーム内に生じる全ての可能なビットパターンと異なるワードを見出す。潜在的(potential)ワードのパフォーマンスの比較のための自然距離(natural metric)は、ワードと、VLCテーブルから導きうるビットストリーム内の全ての可能なパターンの集合と、の間のハミング距離である。正のハミング距離は、ビットストリーム内の全てのビットパターンと異なるワードを意味する。従って、与えられたワード長Rに対しての意図は、長さRの 2^R 個のワードの中から、VLCテーブルから導かれるビットストリーム内に生じる長さRの全ての可能なビットパターンからの最大ハミング距離を有するワードを見出すことである。もちろん、もし長さRの全てのワードが、それらのビットパターンからのハミング距離0を有すれば、Rを増大させなければならない。

【0020】長さRの潜在的再同期ワードと、長さRのビットストリームビットパターンと、の間のハミング距離を見出すためには、ビットパターンの集合を、探索のための3つの部分空間、すなわち、少なくともRの長さを有するコードワードの部分空間 S_1 と、少なくともRの長さの和を有する、コードワードの許容される順序づけられ連結された対の部分空間 S_2 と、少なくとも3つのコードワードの許容される連結であって、該連結の組み込まれた（内部）コードワードのそれぞれがRより小さい長さを有する前記連結の部分空間 S_3 と、に分割する。潜在的再同期ワードrに関する探索は以下のように進められる。

【0021】部分空間 S_1 における第1探索：

(1) 変数Hを、 $L(r)$ であるように初期化する。ただし、 $L()$ は、その引数の長さ（ビット数）であり、Hは、第1探索の終了時において、 r から部分空間 S_1 までのハミング距離となる。

(2) ワード r と、 S_1 内のコードワード c_i^k と、の間のハミング距離を見出すために必要なシフトの総数は、 $L(c_i^k) - L(r) + 1$ であるので、シフトカウンタを $N = L(c_i^k) - L(r) + 1$ に初期化する。

(3)

【外1】

ς

を、 c_i^k のビットNから始まる長さ $L(r)$ の c_i^k のセグメントとして定義する。次に、Hを次式により更新する。

【0022】

【数1】

$$H = \min(H, D(c_i^k, r))$$

ただし、 $D(a, b)$ は、パラメータ a および b の間のハミング距離である。

【0023】 (4) Nを1だけ減少させ、もしNが正ならばステップ(3)へ行く。部分空間 S_1 内のそれぞれのコードワードに対して、以上の(2)から(4)までのステップを繰返した後には、Hは r から S_1 までのハミング距離となり、 H_1 で表される。もちろん、もしHが0になれば、 r は可能な再同期ワードではなく、探索は終了してよい。

【0024】 部分空間 S_2 における第2探索を継続する：

(1) 変数Hを、 $L(r)$ であるように初期化する。Hは、第2探索の終了時において、 r から部分空間 S_2 までのハミング距離となる。

(2) ワード r と、 S_2 内の連結された2つのコードワード $c_i^k + c_j^n$ と、の間のハミング距離を見出すために必要なシフトの総数は、 $L(c_i^k) + L(c_j^n) - L(r) + 1$ であるので、シフトカウンタを $N = L(c_i^k) + L(c_j^n) - L(r) + 1$ に初期化する。

(3)

【外2】

ς

を、ビットNから始まる長さ $L(r)$ の $c_i^k + c_j^n$ のセグメントとして定義する。次に、Hを次式により更新する。

【0025】

【数2】

$$H = \min(H, D(c_i^k, r) + D(c_j^n, r))$$

【0026】 (4) Nを1だけ減少させ、もしNが正ならばステップ(3)へ行く。部分空間 S_2 内のそれぞれの順序づけられたコードワードの対に対して、以上の

(2) から(4)までのステップを繰返した後には、Hは r から S_2 までのハミング距離となり、 H_2 で表される。再び、もしHが0まで減少すれば、 r は可能な再同期ワードではなく、探索は終了してよい。

【0027】 最後に、部分空間 S_3 における第3探索を行う。コピーレントブロックは、VLCテーブルから選択された、 $L(c_q^p)$ が $L(r)$ より小さいコードワード c_q^p として定義される。図3に示されているように、これは中央片であり、これに対して左右へのコードワードが連結される。VLCテーブル内のことごとのコピーレントブロックに対し、以下のように進められる。

【0028】 (1) 変数Hを、 $L(r)$ であるように初期化する。Hは、第3探索の終了時において、 r から部分空間 S_3 までのハミング距離となる。

(2) ワード r と、 S_3 内のコピーレントブロック c_q^p を有する3つまたはそれ以上のコードワードの連結

と、の間のハミング距離を見出すために必要なシフトの総数は、 $L(r) - L(c_q^p) + 1$ であるので、シフ

トカウンタを $N = L(r) - L(c_q^p) + 1$ に初期化する。

(3) r を3つの（空であることも可能な）部分に区分する。それらのうちの、 r_1 は最初の $N-1$ ビットであり、 r_2 は次の $L(c_q^p)$ ビットであり、 r_3 は残りの $L(r) - L(c_q^p) - N+1$ ビットである。

【0029】 (4) 前記コピーレントブロックの両端に許容されるコードワードを反復して連結し、ハミング距離を計算する：

(a) 前記コピーレントブロックの左へのコードワードの許された結合を、その長さが少なくとも $N-1$ になるまで形成し、

【外3】

ς₁

を、この結合の最後の $N-1$ ビットとして定義する。

(b) (a) から得られた前記コピーレントブロック+左方結合、の右へのコードワードの許された結合を、該右方結合の長さが少なくとも $L(r) - L(c_q^p) - N+1$ になるまで形成し、

【外4】

ς₃

を、この右方結合の最初の $L(r) - L(c_q^p) - N+1$ ビットとして定義する。

(c) Hを次式により更新する。

【0030】

【数3】

$$H = \min(H, D(c_1, r_1) + D(c_q^p, r_2) + D(c_3, r_3))$$

【0031】 (d) 全ての許容される左方および右方結合に対し、(a) から(c) までのステップを繰返す。

(e) Nを1だけ減少させ、もしNが正ならばステップ

(3) へ行く。VLCテーブル内のそれぞれのコヒーレントブロックに対して、(2) から (5) までのステップを繰返した後には、HはrとS₃との間のハミング距離となり、H₃で表される。

【0032】このようにして、rと、全ての可能なビットストリームと、の間のハミング距離は、min

(H₁, H₂, H₃)となる。それゆえ、最適の再同期ワードは（もし存在すれば）、正のハミング距離が見出されるまでワード長を増大させて探索することにより見出されうる。ビットストリームまでの1より大きいハミング距離を有する（もっと長い）ワードを探索することは、バーストエラーの性質により役に立たないようである。本探索方法は、S₃においてコヒーレントブロックに焦点を合わせ、可能な再同期ワードの端部にオーバラップする短いコードワードに焦点を合わせないことにより、探索空間を最小化する。再同期ワードを見出すための探索の戦略は、与えられたVLCテーブルに対してそのようなワードが存在するという仮定に依存している。

【0033】ビデオ対象の再同期

対象に基づくビデオのコード化方法は、ビデオを、動き対象と背景対象とに分解するので、フレームのシーケンスは、ビデオ対象のシーケンスの集合として処理され、それぞれの対象に対して1つのシーケンスが存在する。従って、それぞれのフレームは、別個にコード化される対象の集合としてコード化される。デコーダは、デコードされた諸対象からフレームを再構成する。これは、諸対象が多重解像度でコード化されることを可能にし、デコーダは、ある対象を選択して、より良い視覚のために、より高い解像度でデコードしうる。

【0034】対象の形状、内容（テクスチャ）、および動きは、前述のように動き補償を用いて効率的にコード化されうる。また、対象は（いくらかのマクロブロック内において）比較的小さいかもしれないので、第1実施例のslice()の、マクロブロックの單一行への制限は避ける。この圧縮データに対する、実施例のエラー補正は再び、形状、動きベクトル、およびテクスチャデータを区分し、それぞれの区画間に再同期ワードを備える。これは再び、形状、動きベクトル、およびテクスチャデータの、マクロブロックに基づく公知の処理とは対照的なものとなる。このようにして、再同期ワードを、Iフレームに対してはデータの開始点に、また、Pフレームの開始点のほかに、Pフレーム内のことごとくの検出される対象のための下記のアイテムに対するコードのそれぞれの開始点に、導入する。

(i) 形状（例えば、境界輪郭データ）、(ii) 動きベクトルデータ、および (iii) テクスチャデータ (DCT または他の方法が、ウェーブレット (wavelet) のような残余データを圧縮している）。

【0035】さらに、もし制御データ、または他のデータも含まれていれば、このデータも再同期ワードを有し

うる。再同期ワードは、それらが独自のものであること、すなわち、それらが静的テーブルであるVLCテーブル内に存在しないため、それらが、同じ長さのコード化ビットのどのような与えられたシーケンスとも異なること、を特徴とする。例えば、もしPフレームが3つの動き対象を有していたとすれば、シーケンスは次のようなものとなろう。

フレームの初めの再同期ワード

輪郭（形状）再同期ワード

10 第1対象の輪郭データ

動きベクトル再同期ワード

第1対象の動きベクトルデータ

テクスチャ再同期ワード

第1対象のテクスチャデータ

輪郭（形状）再同期ワード

第2対象の輪郭データ

動きベクトル再同期ワード

第2対象の動きベクトルデータ

テクスチャ再同期ワード

20 第2対象のテクスチャデータ

輪郭（形状）再同期ワード

第3対象の輪郭データ

動きベクトル再同期ワード

第3対象の動きベクトルデータ

テクスチャ再同期ワード

第3対象のテクスチャデータ

これらの再同期ワードはまた、デコーダがエラーを検出することを助ける。

【0036】デコーダが、受けたビットストリーム内にエラーを検出した時は、それは最も近い再同期ワードを見出すことを試みる。このようにして、デコーダは、最小のコード化データの損失をもって、可能な最も早い時点において同期を再確立する。

【0037】もし以下の条件のいずれかが観察されれば、デコーダにおいてエラーが検出されうる：

(i) 無効なコードワードが発見される、(ii) デコード中に無効なモードが検出される、(iii) データのデコードされたブロックに再同期ワードが続いていない、

(iv) 動きベクトルがフレームの外部をポイントしている、(v) デコードされたdct値が許容限度外にある、(vi) 境界輪郭が無効である（画像の外側にある）。

【0038】もし境界輪郭データ内にエラーが検出されれば、輪郭は廃棄され、背景の部分とされる。これは、前のフレームの対応する領域が用いられることを意味する。これは、あるひずみを減少させるが、そのわけは、3がしばしばビデオシーケンスにおける時間的相関であるからである。

【0039】もし動きベクトルデータ内にエラーが検出されれば、対象自身の動きベクトルを用い、対象のため

11

の平均動きベクトルが、それぞれのマクロブロックではなく、対象全体に適用される。これは、与えられたフレーム内に大きい空間的相間が存在する事実に基づく。従って、与えられた対象の動きベクトルの大部分は、ほぼ同じである。すなわち、対象のさまざまなマクロブロックに適用される平均動きベクトルは、良い近似となり、視覚的ひずみを顕著に減少させるのに役立つ。もしエラーがテクスチャデータ内に検出されれば、全てのテクスチャデータはゼロにセットされ、デコーダは再同期を試みる。

【0040】ビデオ対象の動きの再同期

再同期のための前述の対象データ区分の明示的な例は、実験的に調査され、コード化のために必要とされる追加ビットの小さいオーバヘッドにより、パフォーマンスの改善を示した。特に、Pタイプの画像に対しても、ちょうど動きベクトルデータおよびテクスチャデータが用いられ、図5がそのビットストリームを示している。それぞれのマクロブロックのための動きベクトルデータは、2つの部分、すなわち、動きベクトルの数と、実際の動きベクトルと、から成る。動きベクトルの数は、1、2、または4のいずれかであり、これは動き補償に対応せず、マクロブロック全体に対して単一動きベクトルが用いられるか、またはマクロブロックを作る4つの8×8ブロックのそれぞれに対して1つの動きベクトルが用いられる。動きベクトルの数は、以下のVLCテーブルによりコード化される。

【0041】

【表1】

| | |
|---|----|
| 0 | 11 |
| 1 | 0 |
| 4 | 10 |

【0042】動きベクトルは、(前のフレームとは異なり)垂直成分に先立ち水平成分によりコード化され、それぞれの成分は、以下のVLCテーブルによりコード化される。ただし、該テーブルにおいて、sは、+エントリに対しては0、-エントリに対しては1、に等しい。

【0043】

【表2】

10

20

30

40

50

12

| | |
|-------|------------------|
| ±16 | 0000 0000 0010 s |
| ±15.5 | 0000 0000 0011 s |
| ±15 | 0000 0000 010s |
| ±14.5 | 0000 0000 011s |
| ±14 | 0000 0000 100s |
| ±13.5 | 0000 0000 101s |
| ±13 | 0000 0000 110s |
| ±12.5 | 0000 0000 111s |
| ±12 | 0000 0001 00s |
| ±11.5 | 0000 0001 01s |
| ±11 | 0000 0001 10s |
| ±10.5 | 0000 0001 11s |
| ±10 | 0000 0010 00s |
| ±9.5 | 0000 0010 01s |
| ±9 | 0000 0010 10s |
| ±8.5 | 0000 0010 11s |
| ±8 | 0000 0011 00s |
| ±7.5 | 0000 0011 01s |
| ±7 | 0000 0011 10s |
| ±6.5 | 0000 0011 11s |
| ±6 | 0000 0100 00s |
| ±5.5 | 0000 0100 01s |
| ±5 | 0000 0100 1s |
| ±4.5 | 0000 0101 0s |
| ±4 | 0000 0101 1s |
| ±3.5 | 0000 011s |
| ±3 | 0000 100s |
| ±2.5 | 0000 101s |
| ±2 | 0000 11s |
| ±1.5 | 0001 s |
| ±1 | 001s |
| ±0.5 | 01s |
| 0 | 1 |

【0044】このようにして、許容されるビットストリームは、前記のベクトル数のVLCテーブルからの前のエントリに依存するこのVLCテーブルからの2つまたは8つの連続するエントリを有する。ビットストリームは、フレームの予測タイプ(例えば、I、P、またはB)と、テクスチャデータのための量子化因子と、のような対象アイテムのためのある固定長コードと、マクロブロックデータをパッケージ化するため、テクスチャデータに続く17ビットの再同期マーカ00000000000000001と、を有し、動きベクトルデータおよび再同期マーカのためのこれら2つのVLCテーブルと共に前節の実施例の方法を用いた探索は、ほぼ10個の可能な最小長(17ビット)の動き再同期ワードを与えた。特定の動き同期ワード(10100000000000001)が選択され、ランダムビットエラーと、パケット損失エラーと、バーストエラーとにより、ビットストリームを劣化させることによって、雑音性チャネルを経ての送信のシミュレーションに用いられた。図4は、マクロブロックシーケンスによるマクロブロック内の通常の動きベクトルおよびテクスチャデータと比較した時の、エラー許容性(resiliency)ビットストリーム(動きベクトルデータと、テクスチャデータとを、間に動き再同期ワードを置くことにより区分している)のパフォーマンスを示す。シミュレーションにおいて、ビット誤り率は10⁻²であり、バ

ースト長は1msであった。図4は、ピークSN比(P SNR)をフレーム番号の関数として示す。動きベクトルとテクスチャデータとの動き再同期ワードによる区分は、2dBより大きい利得を生じる。

【0045】ビデオ対象の形状および動きの再同期

多重対象のための形状データ、動きベクトルデータ、およびテクスチャデータを含むビットストリームにおける実施例のコード化は、諸対象のデータを分離する再同期マーカをもつパケットを有する。図5からわかるように、再同期マーカのそれぞれの対の間ににおいて、单一対象のマクロブロックの集合のためのデータは、形状データと、動きベクトルデータと、テクスチャデータと、に区分され、形状データと、動きベクトルデータと、の間には、形状再同期ワードがあり、動きベクトルデータと、テクスチャデータと、の間には、動き再同期ワードがある。動きベクトルデータは、再び、動きベクトルの数と、異なる(differentia1)動きベクトル成分とを含む。形状データは、対象識別データと、形状コードとを含む。

【0046】対象のサイズ(マクロブロックの数)は、大きく変動しうるので、单一のパケットは、例えば、第1対象のマクロブロックの最後の部分と、第2対象のマクロブロックの全てと、第3対象のマクロブロックの最初の部分と、を含みうる。この場合、再同期マーカは、3つの対象のデータの集合を分離し、動き再同期ワードは、それぞれの対象のためのデータを、形状、動き、およびテクスチャデータに区分する境界となる。

【0047】別のシナリオにおいては、それぞれの個々の対象を別々にパケット化することが好ましい。この場合には、1つの対象のみの動きおよびテクスチャデータが、2つの連続する再同期マーカの間に現れる。図6は、1フレーム内の2つの対象のマクロブロックを通る走査を示し、図7は、ビットストリームのシンタックスを示す。このスキームにおいても、2つの再同期マーカの間に形状および動きが現れる。このアプローチの利点は、それぞれの対象に属するデータが、ビットスキーム内において別個にパケット化されることである。

【0048】一般に、再同期マーカは、固定した間隔でビットストリーム内に挿入されうるので、対象のデータは1つより多くの形状-動き-テクスチャのグループに分割されうる。例えば、48kbpsのような低ビットレートおよび高圧縮の場合は、再同期マーカは764ビット毎に用いられる。より高いレートにおいては、もっと頻繁でない再同期マーカが用いられる。

【0049】形状再同期ワードおよび動き再同期ワードは、それぞれ実施例の探索方法により発生せしめられる。前述のように、再同期ワードは、たとえもしデータのいくつかが廃棄されなければならなくても、エラー検出を助け、部分的なデータの使用を可能にする。例えば、形状データおよび動きベクトルデータは、テクスチ

ヤデータなしでも使用されうる。

【0050】改善されたエラー許容性のための1つの可能性は、形状再同期ワードおよび動き再同期ワードと共にVLCテーブル内に可逆コード(コードワードが対称である)を使用することにより得られる。これは、検出されたエラーを局部化する利点を有する。デコーダがエラーを検出した時は、デコーダは次の再同期マーカまでジャンプし、前に検出されたエラーに向かって後方へデコードする。VLCの使用は、しばしば、デコーダがエラーの位置を通過した後にのみエラー検出を可能にするので、後方へのデコーディングは、順方向エラー検出の位置を通過するまで多分エラーを検出しない。図7を参照されたい。この場合は、順方向デコードされたエラーと、後方検出されたエラーと、の間のデータを廃棄する。これは、そのエラーからのデータ再生の量を最大化する。

【0051】以上の説明に関して更に以下の項を開示する。

20 (1) (a) 対象のピクセルの複数のグループのための動きデータおよびテクスチャデータを発生するステップと、(b) 前記動きデータを集約するステップと、(c) 前記テクスチャデータを集約するステップと、(d) 前記集約された動きデータと、前記集約されたテクスチャデータとの間に、再同期ワードを挿入するステップと、を含む、ビデオ情報をパケット化する方法

【0052】(2) 前記グループのピクセルが、ビデオ情報のフレームの16×16マクロブロック内にある前記対象の前記ピクセルである、第1項に記載の方法。

30 (3) (a) 形状データを集約するステップと、(b) 前記動きデータおよび前記テクスチャデータから前記形状データを区別するために第2再同期ワードを挿入するステップと、をさらに含む、第1項に記載の方法。

【0053】(4) 動き補償されたビデオビットストリームシンタックスであって、(a) ビットストリーム内の第1グループの連続ビットであって、少なくとも2つの動きベクトルをコード化した該第1グループの連続ビットと、(b) 前記ビットストリーム内の前記第1グループのビットに続く第2グループの連続ビットであって、再同期ワードを形成する前記第2グループの連続ビットと、(c) 前記第2グループに続く第3グループの連続ビットであって、前記動きベクトルに関連するテクスチャデータをコード化した前記第3グループの連続ビットと、を含む、前記動き補償されたビデオビットストリームシンタックス。

【0054】(5) 可変長コードの集合から独自のワードを発生する方法であって、(a) 可変長コードワードの集合を発生するステップと、(b) 正の整数Rに対し、前記コードワードのシーケンス内に生じうる長さRの全てのビットパターンの空間を、第1、第2、および第3部分空間に区分するステップであって、前記第1部

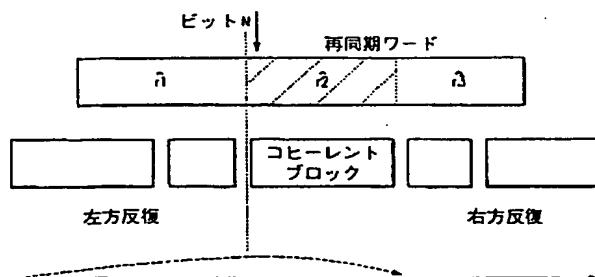
分空間が、少なくともRの長さのコードワード内の、全ての長さRのシーケンスであり、前記第2部分空間が、少なくともRの長さの和を有する、2つの前記コードワードの許容される連結内の、全ての長さRのシーケンスであり、前記第3部分空間が、少なくともRの長さの和を有する、3つまたはそれ以上の前記コードワードの許容される連結内の、全ての長さRのシーケンスであり、前記3つまたはそれ以上のコードワードの内部コードワードの長さがRより小さい長さである、前記区分するステップと、(c) 長さNの潜在的再同期ワードを発生するステップと、(d) 前記潜在的再同期ワードから前記第1部分空間までの距離を計算するステップと、(e) 前記潜在的再同期ワードから前記第1部分空間までの距離が0より大きい時に、前記潜在的再同期ワードから前記第2部分空間までの距離を計算するステップと、

(f) 前記潜在的再同期ワードから前記第2部分空間までの距離が0より大きい時に、前記潜在的再同期ワードから前記第3部分空間までの距離を計算するステップと、(g) 前記潜在的再同期ワードから前記第3部分空間までの距離が0より大きい時に、前記潜在的再同期ワードを再同期ワードとして選択するステップと、(h) 前記第1、第2、または第3部分空間までの前記距離が0である時に、長さNのもう1つの潜在的再同期ワードを発生し、ステップ(c)から始まる以上のステップを繰返すステップと、(i) 長さNの全ての潜在的再同期ワードにおける、前記第1、第2、または第3部分空間までの前記距離が0である時に、NをN+1により置換

【図1】

| スライス 再同期 | MB 番号 | 0 | MV _n | MV _{n+1} | ... | MV _{n+k} | 動き 再同期 | DCT _n | ... | DCT _{n+k} |
|-------------|----------|---|-----------------|-------------------|-----|-------------------|-----------|------------------|-----|--------------------|
|-------------|----------|---|-----------------|-------------------|-----|-------------------|-----------|------------------|-----|--------------------|

【図3】



【図5】

| | | | | | | | | | | | | | |
|------------|--------------|-----------|--------------|-----------|--------------------|-----|------------|-----------|--------------|-----------|--------------------|------------|-----|
| 再同期 マーカ | 対象1 形状データ | 形状 再同期 | 対象1 動きデータ | 動き 再同期 | 対象1 テクスト データ | 再同期 | 対象2 マーカ | 形状 再同期 | 対象2 動きデータ | 動き 再同期 | 対象2 テクスト データ | 再同期 マーカ | ... |
|------------|--------------|-----------|--------------|-----------|--------------------|-----|------------|-----------|--------------|-----------|--------------------|------------|-----|

し、ステップ(c)から始まる以上のステップを繰返すステップと、を含む、前記方法。

【0055】(6) ビットストリームにおいて、動きベクトルが集約され且つ対応するテクスチャデータから再同期ワードにより分離されている、前記ビットストリームを有する、(マクロ) ブロックレベルの動き補償により圧縮されたビデオ、および、可変長コードテーブルから再同期ワードを発生する方法であって、該発生せしめられた再同期ワードを用いて、該再同期ワードに隣接する動きベクトルまたはテクスチャデータがコード化される、前記方法。

【0056】(関連出願に対するクロスリファレンス) 本出願は、1996年10月25日付出願の特許出願第08/739,111号に関連する。

【図面の簡単な説明】

【図1】第1実施例の再同期を有する、ビットストリームパケットシンタックスを示す。

【図2】再同期を有する、公知のビットストリームパケットシンタックスを示す。

【図3】再同期ワード探索を示す。

【図4】実験結果を示す。

【図5】もう1つのビットストリームシンタックスを示す。

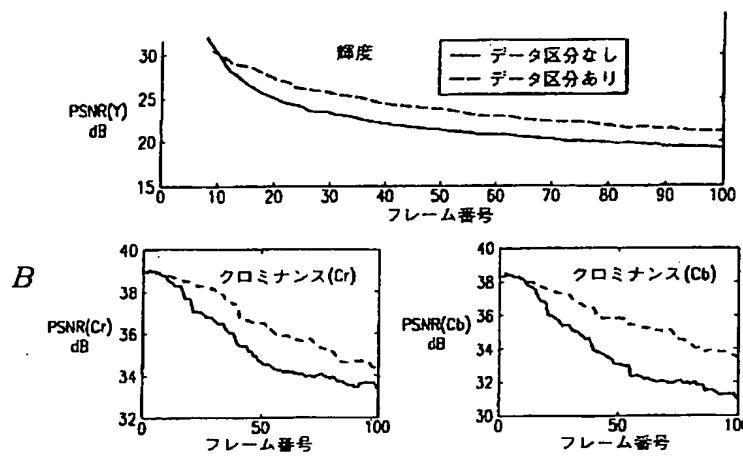
【図6】対象走査を示す。

【図7】もう1つのビットストリームシンタックスを示す。

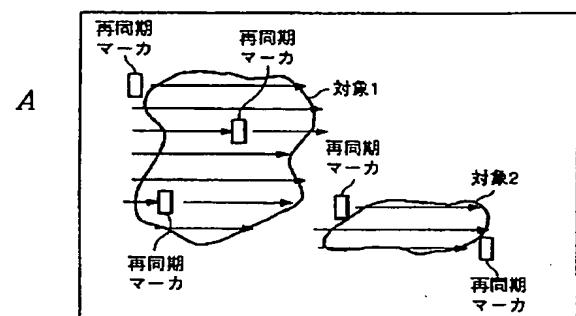
【図2】

| スライス 再同期 | MB 番号 | 0 | MV _n | DCT _n | MV _{n+1} | DCT _{n+1} | ... | MV _{n+k} | DCT _{n+k} |
|-------------|----------|---|-----------------|------------------|-------------------|--------------------|-----|-------------------|--------------------|
|-------------|----------|---|-----------------|------------------|-------------------|--------------------|-----|-------------------|--------------------|

【図4】



【図6】



【図7】

| | | | | | | | | | |
|--------|----------|-------|----------|-------|-------------|--------|----------|-------|----------|
| 再同期マーカ | 対象1形状データ | 形状再同期 | 対象1動きデータ | 動き再同期 | 対象1テクスチャデータ | 再同期マーカ | 対象1形状データ | 形状再同期 | 対象1動きデータ |
|--------|----------|-------|----------|-------|-------------|--------|----------|-------|----------|

| | | | | | | | | | |
|-------|-------------|--------|----------|-------|----------|-------|-------------|----------|--------|
| 動き再同期 | 対象1テクスチャデータ | 再同期マーカ | 対象2形状データ | 形状再同期 | 対象2動きデータ | 動き再同期 | 対象2テクスチャデータ | テクスチャ再同期 | 再同期マーカ |
|-------|-------------|--------|----------|-------|----------|-------|-------------|----------|--------|